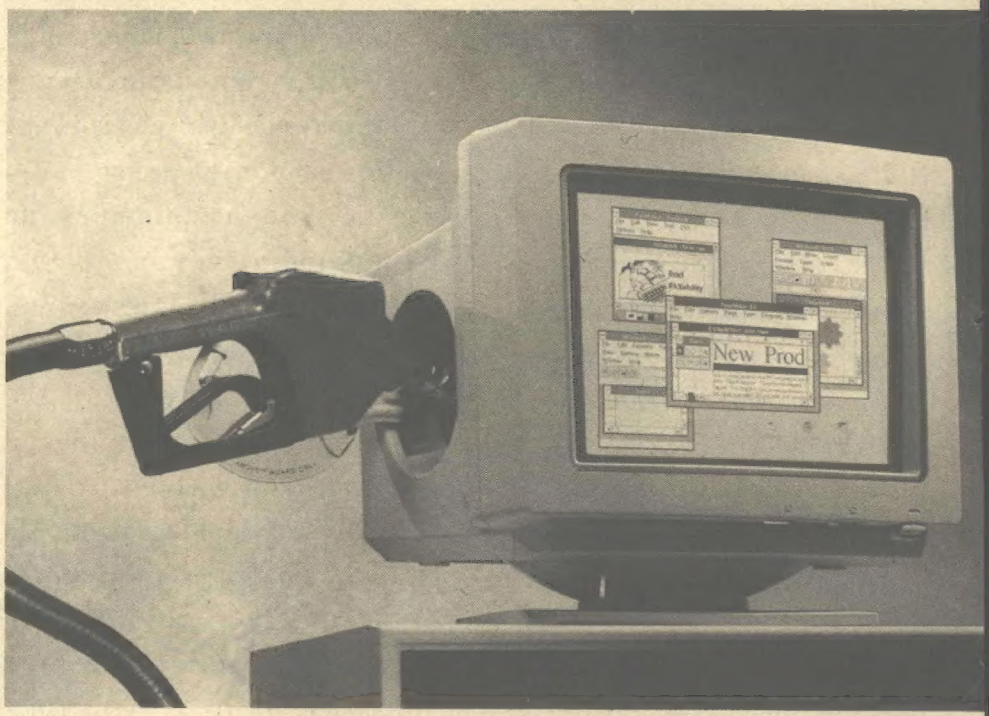
ศรันย์ ไมตรีเวช

# ปัญหาชวนตื้อ สำหรับผู้เริ่มเล่น Visual Basic ตอนที่ 4

***แนะนำวิธีการใช้ โปรแกรม Database Driver ตัวแรกของวิชวลเบสิกคือ Q+E Multilink/VB***




สมัยที่วิชวลเบสิกออกตลาดใหม่ ๆ นั้น จุดประสงค์โดยทั่วไปคือต้องการใช้มันเป็นโปรโตไทพ์ ด้วยเหตุที่เขียนโค้ดได้ง่าย ว่ากันแบบเขียนทิ้งเขียนขว้าง ไม่ได้ต้องการนำมาประยุกต์ใช้กับงานจริงแต่อย่างใด เหตุผลก็คือมันช้า...(โปรดลากเสียงให้ยาวออกไปมาก ๆ เพราะมันช้าแบบผิดวิสัยเกินจะยอมรับ)

แต่ทุกวันนี้มันต่างกับเมื่อวานจากหน้ามือเป็นหลังมือ ความเร็วในการทำงานของเครื่องมือสร้างแอพพลิเคชันตัวนี้ทวีตัวขึ้นอย่างมหัศจรรย์ ใครเคยเห็นความเร็วของ VB 1.0 มาก่อนจะต้องพบกับความงงงันเหลือจะกล่าว คือเว้นแต่งานที่มีรูปใหญ่ ๆ เคลื่อนไหวได้และโค้ดที่โปรแกรมเมอร์ประยุกต์ใช้คำสั่ง/คอนโทรลไม่มีประสิทธิภาพแทบไม่มีแอพพลิเคชันที่สร้างขึ้นด้วยวิชวลเบสิกตัวไหนถูกบ่นว่าช้าเลย

แน่นอน เมื่อมันถูกมองว่าทำ "งานจริง" ได้ สิ่งที่ตามมาเป็นลูกโซ่คือเสียงเรียกร้องจากผู้ใช้ดาต้าเบสทั้งหลาย โฉมหน้าของการโค้ดเพื่อติดต่อกับฐานข้อมูลพลิกเปลี่ยนไปอย่างสิ้นเชิง คุณไม่ต้องประดิษฐ์ประดอยทุกสิ่งทุกอย่างเองเหมือนแต่ก่อน ไม่ต้องมีความรู้วิธีติดต่อกับดาต้าเบสอย่างลึกซึ้ง คุณก็สามารถเรียกดูข้อมูลได้ง่ายราวกับฝันไป

ยกตัวอย่างง่าย ๆ เชื่อไหมว่าคุณไม่ต้องเขียนโค้ดแม้แต่บรรทัดเดียว คุณก็สามารถดึงข้อมูลขึ้นมาได้หนึ่งเรคอร์ดด้วยการกดปุ่มเพียงหนึ่งครั้ง

สำหรับตอนนี้ผมขอแนะนำดาต้าเบสไดรเวอร์คือ Q+E Multilink/VB เป็นตัวแรก เนื่องจากเห็นว่าเป็นที่รู้จักในการใช้งานจริงมากกว่าเพื่อน และในตอนต่อไปถึงจะแนะนำให้รู้จักกับดาต้าเบสไดรเวอร์ของวิชวลเบสิกเอง (รวมทั้ง TRUEGRID.VBX ซึ่งจะทำให้การดำเนินงานกับข้อมูลในรูปของเทเบิลเป็นไปได้เมื่อนำมาประกอบกับ Data Control) คุณจะได้เปรียบเทียบความแตกต่างทั้งในด้านความสามารถและวิธีการ ซึ่งแต่ละฝ่ายก็มีทั้งข้อดีและข้อเสียอยู่ในตัว

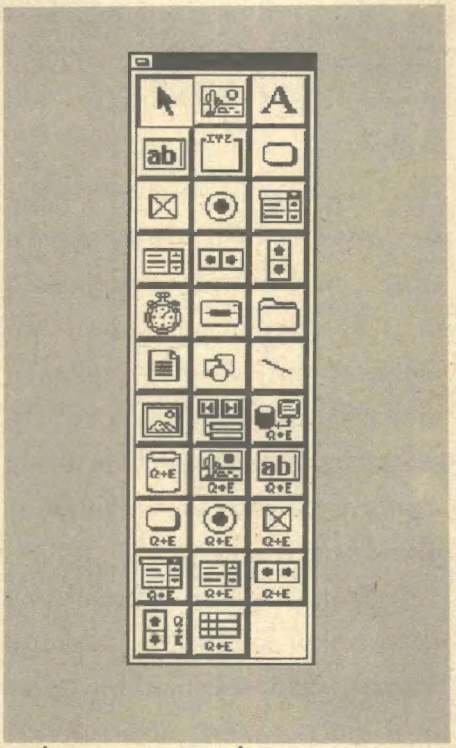
## จะเริ่มใช้ Q+E Multilink/ VB กันอย่างไร

เนื่องจากเนื้อที่มีจำกัด ผมจะไม่ร่ายรำสาธยายอะไรให้เยิ่นเย้อ มาถึงก็ตะลุมบอนกันเป็นข้อ ๆ เลยดีกว่า ที่นี่เราหนักกันเฉพาะ How ครับ ส่วนเรื่อง What นั้นก็มีพอแค่จิ้มลิ้ม ใครอยากได้รายละเอียดมากกว่านี้ก็หาอ่านเอาเองจากคู่มือ

1. ก่อนอื่นคุณต้องมีไฟล์ QELINK.VBX อยู่ใน \WINDOWS\SYSTEM ไฟล์นี้ไฟล์เดียวอัดคอนโทรลที่จำเป็นในการติดต่อกับดาต้าเบสมาให้คุณครบ อย่าแปลกใจเมื่อโหลดแค่ไฟล์เดียวได้ไอคอนของคอนโทรลใหม่ขึ้นมาในทูลส์-บ็อกซ์ตั้งสิบสองตัว...มันเป็นเรื่องของสินค้าราคาเรือนหมื่นน่ะครับ ให้จัดการโหลดมันขึ้นมารวมในโปรเจ็กต์เหมือนคอนโทรลเสริมตัวหนึ่ง (โหลดด้วยเมนู File...Load หรือปุ่ม Alt+FD)

2. จากนั้นคุณควรจะมีไฟล์ QELINK.BAS ไว้ในไดเรคทอรีปัจจุบันที่กำลังใช้สร้างแอพพลิเคชันติดต่อกับดาต้าเบส สำหรับไฟล์นี้จะอยู่ที่ไหนไม่สำคัญ และก็ไม่จำเป็นต้องเป็นชื่อนี้ด้วย เพราะหน้าที่หลักคือดีเคลร์ฟังก์ชันสำคัญที่ใช้ในการติดต่อกับดาต้าเบส คุณอาจตัดแต่งเสียใหม่ก็ได้ เพราะมันก็คือไฟล์โมดูลดี ๆ นี่เอง ไม่ใช่ไฟล์ลึกลับมหัศจรรย์ที่โปรแกรมเมอร์ทั่วไปแตะต้องไม่ได้ ให้จัดการโหลดไฟล์นี้ขึ้นมาอยู่ในโปรเจ็กต์เช่นเดียวกันกับ QELINK.VBX

ตอนแรกคุณอาจหาไม่เจอที่ C:\QELINK ก็ลองไปสืบหาดูในสับไดเรคทอรีย่อยที่เป็นตัวอย่างเช่น C:\QELINK\DAYCARE เป็นต้น



รูปที่ 1 กลุ่มคอนโทรลซึ่งปรากฏให้เห็นในทูลส์บ็อกซ์หลังจากโหลดไฟล์ QELINK.VBX ขึ้นมาเพียงไฟล์เดียว

3. อีกขั้นก่อนลงมือสร้างดาต้าเบสคอนโทรล คุณควรจะดีเคลร์ตัวแปรสำคัญคือ res% (ย่อมาจากคำว่า result_code%) ไว้ที่โมดูล (ฟอร์มนามสกุล .bas) หรือจะดีเคลร์ไว้ที่ Declaration Section ของฟอร์มปัจจุบันก็ได้หากแอพพลิเคชันตัวนั้นมีแค่ฟอร์มเดียว หรือแม้แต่จะให้มันเป็นตัวแปรโลคอลในโพรซีเดอร์ไดโพรซีเดอร์หนึ่งก็ยังไหว เพราะมันมีไว้รับค่ารีเทอร์นจากฟังก์ชันดาต้าเบสเท่านั้น ไม่ได้เอาไปใช้งานอะไรอย่างอื่น อีกทั้งเมื่อตั้งชื่อจริง ๆ ก็ไม่จำเป็นต้องเป็น res% เสมอไป จะเป็น x, y, z อะไรก็ได้ ขอให้เป็นตัวแปร Integer ก็แล้วกัน ใช้ได้ทั้งนั้น (ทว่าในทางปฏิบัติมักใช้ res% กันนะครับ เพราะชินตาเป็นที่หมายรู้ได้ดีว่ากำลังจะใช้มันเรียกฟังก์ชันดาต้าเบส)

สรุปแล้วตอนนี้คุณเตรียมการณ์สำคัญที่สุดไว้สามข้อเพื่อรองรับการเจรจาระหว่างวิชวลเบสิกกับดาต้าเบส นั่นคือโหลดไฟล์ QELINK.VBX, โหลดไฟล์ QELINK.BAS และทำการดีแคลร์ตัวแปรสำคัญเพื่อรับค่ารีเทอร์นจากฟังก์ชันติดต่อกับดาต้าเบส นั่นคือ res%

ต่อจากนั้นก็ถึงเวลาทำความรู้จักกับคอนโทรลหลักกันเสียที คอนโทรลตัวนี้มีชื่อตามหน้าที่ของมันครับ นั่นคือ Connect-ผู้เชื่อมสัมพันธภาพระหว่างแอพพลิเคชันของวิชวลเบสิกกับดาต้าเบส ชื่อจำง่ายเพราะตั้งตามหน้าที่ของมันอยู่แล้ว

ขอให้ดูรูปที่ 2 กันก่อน จะเห็นได้ว่ามันไม่มีอะไรพิสดารในรูปแบบเลยแม้แต่น้อย

ดับเบิลคลิกไอคอน แล้วก็จะได้เจ้านี่ขึ้นมาบนฟอร์ม คุณไม่ต้องไปพะวงกับตำแหน่งที่วางอันเหมาะสม รวมทั้งไม่ต้องไปวุ่นวายเซตค่า Visible ให้เป็น False เนื่องจากมันจะล่องหนไปเองเมื่อแอพพลิเคชันถูกรัน ในขณะ Design Time มันก็แค่โชว์ตัวให้เห็นว่ามันมาแล้วเท่านั้น

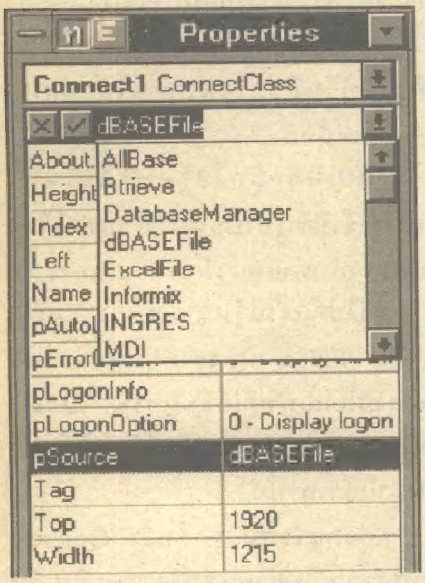
ขอกล่าวสั้น ๆ ว่าเจ้า Connect ตัวนี้ทำตัวเป็นล่ามสารพัดภาษาที่น่าทึ่ง นั่นคือมันรู้จักดาต้าเบสทุกชนิด ขอเพียงระบุมาเท่านั้นว่าจะติดต่อกับใคร

รูปที่ 2 โฉมหน้าของ Connect เมื่อสร้างขึ้นมาบนฟอร์มทีแรก

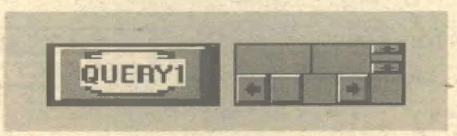
และการระบุชนิดดาต้าเบสก็เป็นหน้าที่อันหลีกเลี่ยงไม่ได้ของคุณเสียด้วยสิครับ โปรดจำไว้เลย เมื่อสร้าง Connect แล้วให้กำหนดชนิดของดาต้าเบสทันที (ถ้าคิดว่าไม่ลืมจะไปเขียนในโค้ดก็ได้) โดยใช้พรอพเพอร์ตี้ชื่อ pSource ซึ่งก็หวังว่าคงเห็นชัดเจนดีแล้วในรูปที่ 3

การให้ค่าชนิดดาต้าเบสแก่ pSource นั้นเป็นการเพียงพอแล้วสำหรับคอนโทรล Connect ในที่นี้ผมขอยก dBASE Files มาใช้สาธิตเพราะเห็นว่ามันง่ายดีและเป็นที่รู้จักโดยทั่วอยู่แล้ว

ลำดับต่อไปมาทำความรู้จักกับ Query และ QGrid กันเลย หน้าตาพวกมันเป็นดังรูปที่ 4




**รูปที่ 3 ดาต้าเบสที่ Connect รู้จัก ข้างล่างยังมีต่ออีกเยอะ เป็นต้นว่า Oracle, NetWareSQL**




**รูปที่ 4 เหล่าคอนโทรลหลักที่ผู้จะนำมาซึ่งข้อมูลทั้งหลาย**

ด้านซ้ายมือนั่นชื่อบอกอยู่แล้วว่าเป็น Query ส่วนด้านขวามือคือ QGrid ในขณะยังไม่ได้ขยายตัวอวดโฉมหน้าที่แท้จริง

ขอพูดกระซับ ๆ เกี่ยวกับตัวเรียกข้อมูลทั้งสองนี้เสียหน่อย สำหรับ Query นั้น เมื่อมันรู้จักกับชื่อเทเบิลใดแล้ว ก็สามารถกวาดเอาข้อมูลมาได้ทีเดียวเรียบ ทุกฟิลด์ ทุกเรคอร์ด แต่น่าเสียดายที่แสดงข้อมูลด้วยตัวเองไม่ได้เลย เปรียบไปก็เหมือนเครื่องฉายหนังต้องยืมฉากรับแสงสีจากวัตถุอื่นตัวเองเป็นแต่ส่งอย่างเดียว

ส่วน QGrid นั้นเหนือชั้นขึ้นมากว่านั้นหน่อย คือเป็นทั้งตัวเรียกข้อมูลทั้งหมดมาเก็บไว้ในตัวได้เหมือน Query มิหนำซ้ำยังแสดงข้อมูลได้ด้วยตัวเองโดยไม่จำเป็นต้องยืมคอนโทรลตัวอื่นมาเป็นสถานีแสดงแทน จะให้แสดงฟิลด์ไหนบ้างคุณสามารถระบุได้ตามใจชอบ และเมื่อเลือกฟิลด์แล้ว ถ้าไม่กำหนดด้วยเงื่อนไขอื่น มันจะนำทุกเรคอร์ดของฟิลด์เหล่านั้นขึ้นมาแสดง ฟังดูเข้าท่าดีไหมครับ

เมื่อขยายมันออกกว้างคุณจะเห็นตัวจริงของมันเป็นดังรูปที่ 5

เห็นอย่างนี้คุณคงมักจะเผลอนึกว่ามันเป็นคอนโทรลแบบเดียวกับ Grid สามัญ แต่จริง ๆ แล้วไม่ใช่ โปรดระลึกว่ามันเป็นตัวเรียกข้อมูลเหมือนกับ Query เปรียบไปก็คล้ายโทรทัศน์ที่สามารถรับสัญญาณคลื่นและแสดงภาพได้ในตัวเอง ไม่ต้องยืมวัตถุอื่นมารองรับภาพให้วุ่นวายอีก

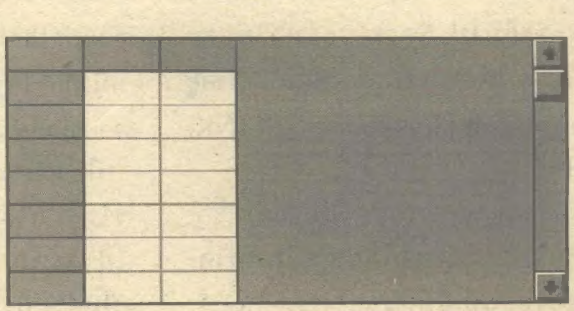
## การระบุชื่อ Connect

ทั้ง Query และ QGrid นั้นมีพรอพเพอร์ตี้สำคัญที่สุดเหมือนกัน นั่นคือ pConnectName อันนี้แหละครับที่จะเป็นตัว "เสียบแจ็ค" เข้าเครือข่ายวงจรปิดให้เกิดความสามารถในการทำงานขึ้นมาได้

คงไม่ต้องเดาให้วุ่นวายว่าการเสียบแจ็คนั้นหมายถึงอะไร ชื่อพรอพเพอร์ตี้มันฟ้องตัวเองตามเคย มันก็คือการเชื่อมตัวเองกับ Connect นั่นเองครับ ถ้ายัง "อันปลั๊ก" อยู่ก็หมดความสามารถจะทำงานได้ เพราะพูดภาษาดาต้าเบสเองไม่เป็น

การใส่ค่าพรอพเพอร์ตี้นั้นก็เหมือนกับการใส่ค่าพรอพเพอร์ตี้ประเภทเดียวกันทั่วไป คือคุณจะพิมพ์เองโดยตรงหรือจะเลือกเอาจากช่องลำดับรายการก็ได้ สิ่งสำคัญคือต้องมี Connect อยู่แล้วหนึ่งตัวในฟอร์มนั้น การระบุชื่อ Connect จึงจะเกิดขึ้นได้

มีกรณีพิเศษตรงที่คุณอยากใช้ Connect ตัวเดียวจากฟอร์มหลัก เรียกว่าสร้าง Connect ขึ้นมาตัวเดียวทำ



**รูปที่ 5 โฉมหน้าของ QGrid หลังจากยืดออกมาได้ขนาดพอควร**

หน้าที่เป็นแม่แหล่งควบคุม Query และ QGrid ได้ทั่วทุกฟอร์มในแอพพลิเคชัน ก็อาจทำได้โดยการอ้างชื่อ Connect ข้ามฟอร์ม

ยกตัวอย่างเช่นสมมติว่าปัจจุบันคุณอยู่ที่ Form2 แต่ Connect1 เขาอยู่ที่ Form1 เมื่อจะระบุชื่อลงไปใน pConnectName ก็ให้เอา Form1 อันเป็นที่อยู่ขึ้นหน้าแล้วต่อด้วย .Connect1 อันเป็นชื่อเสียงเรียงนามของมันตามหลัง รวมกันเป็น Form1.Connect1 ถ้าไม่เห็นจุดที่คั่นระหว่าง Form1 กับ Connect1 ก็ให้กลับไปเห็นเสียใหม่นะครับ ขาดไม่ได้เด็ดขาด

## รู้จักกับ Query Builder

เมื่อคุณมีทั้ง Connect และ Query (หรือ QGrid) อยู่บนฟอร์มเรียบร้อย ต่อไปที่จะต้องทำคือกำหนดรายละเอียดการเรียกข้อมูลต่าง ๆ อันประกอบด้วย

1. การเลือกเทเบิล
2. การเลือกฟิลด์
3. การตั้งเงื่อนไข
4. การกำหนดลำดับข้อมูล

ซึ่งการกำหนดรายละเอียดทั้งหมดนี้หนีไม่พ้นพรอพเพอร์ตีส์สำคัญสี่ตัวคือ

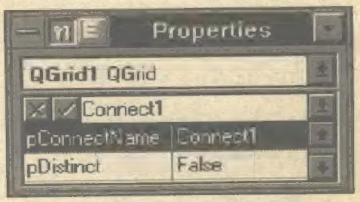
1. pTables (เลือกเทเบิล)
2. pQueryExpr (เลือกฟิลด์)
3. pWhere (ตั้งเงื่อนไข+กำหนดลำดับข้อมูล)
4. pDistinct (ให้เรียกเฉพาะเรคอร์ดที่ไม่ซ้ำ)

ขึ้นชื่อว่าเป็นพรอพเพอร์ตีส์แล้วคุณสามารถให้ค่ามันได้ตรง ๆ ด้วยการพิมพ์ที่หน้าต่างพรอพเพอร์ตีส์เสมอ แต่ในกรณีของการกำหนดรายละเอียดเรียกข้อมูลนั้นมันมีองค์ประกอบอะไรต่อมิอะไรอยู่เยอะ ใหม่ ๆ คุณยังอาจขาดความเข้าใจครบถ้วน ถ้าพิมพ์เองที่หน้าต่างพรอพเพอร์ตีส์อาจทำผิดทำถูก เพราะฉะนั้นเขาจึงช่วยคุณด้วยการให้หน้าต่างเฉพาะกิจขึ้นมาทั้งการเลือกเทเบิล การเลือกฟิลด์ การตั้งเงื่อนไข และการกำหนดลำดับข้อมูล ทั้งหมดนี้รวมกันเรียกว่า Query Builder

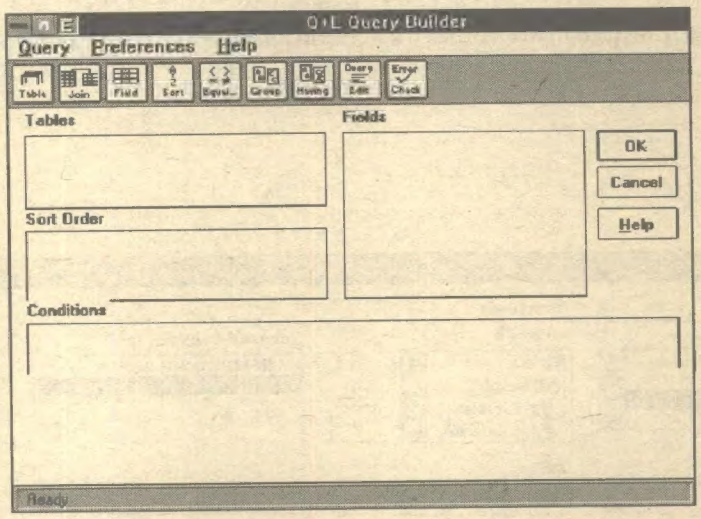
ตัว Query Builder นี้มีศูนย์กลางอันเปรียบเสมือนแหล่งแสดงรายละเอียดการเลือกทั้งหมด และยังเป็น "เมนูพิเศษ" เพื่อเข้าไปยังหน้าต่างย่อยอันประกอบด้วย Tables, Fields, Conditions, และ Sort Order พูดง่าย ๆ ว่ามันทำหน้าที่ช่วยคุณกำหนดรายละเอียดการเรียกข้อมูลได้อย่างครบถ้วน ขอให้ดูหน้าตามันในรูปที่ 7

การเรียก Query Builder ขึ้นมาใช้นั้น ทำได้ตั้งแต่คุณมี Connect และ Query (หรือ QGrid) อยู่บนฟอร์ม โดยตอนแรกคุณจะไม่เห็นหน้าต่าง Query Builder อย่างในรูปที่ 7 หรอกครับ คุณต้องเลือกเข้าไปในหน้าต่างเฉพาะกิจบานใดบานหนึ่ง วิธีคือคลิ๊กปุ่ม ... ข้างช่องใส่ค่า หรืออาจดับเบิลคลิ๊กตรงตัวพรอพเพอร์ตีส์นั้น ๆ เช่น pTables เป็นต้น (ส่วนตัวผมถนัดใช้วิธีหลังมากกว่า และขอกล่าวอ้างใช้วิธีนี้ต่อไป) ถ้าหากว่ายังไม่มีการเลือกเทเบิลแล้วคุณดับเบิลคลิ๊กที่พรอพเพอร์ตีส์ตัวอื่นอย่าง pQueryExpr หรือ pWhere มันก็จะให้คุณระบุชื่อเทเบิลก่อนด้วยการเรียกหน้าต่าง Tables มาก่อน เนื่องจากหากปราศจากเทเบิล ก็ไม่รู้ว่าเราจะเลือกฟิลด์และตั้งเงื่อนไขกันได้อย่างไร

เรามาพูดถึงการกำหนดเรียกข้อมูลกันเป็นลำดับกันไปเลยครับ




**รูปที่ 6 พรอพเพอร์ตี้ "เสียบแจ็ค" ที่ชื่อ pConnectName**




**รูปที่ 7 หน้าต่าง Query Builder ผู้สร้างแสดงรายละเอียดการเรียกข้อมูลสมบูรณ์แบบ**

## การเลือกเทเบิล

ทั้ง Query และ QGrid ใช้พรอพเพอร์ตี้ pTables ในการเลือกเทเบิลเหมือน ๆ กัน คุณจะพิมพ์ชื่อเทเบิลลงไปตรง ๆ ก็ได้ แต่ตอนนี้ผมจะแนะนำให้รู้จักกับหน้าต่าง Tables เสียก่อน มาดูกันครับว่าเราจะเรียกมันขึ้นมาได้อย่างไร

### วิธีเข้าหน้าต่าง Tables

- ถ้าอยู่ที่หน้าต่างพรอพเพอร์ตี้ส์ ให้ดับเบิลคลิ๊กบรรทัด pTables
- ถ้าอยู่ที่หน้าต่าง Query Builder ให้คลิ๊กลงในบริเวณที่ว่างของช่อง Tables

ขอซักซ้อมกันอีกทีนะครับ จริง ๆ แล้วเราเข้ามาใน Query Builder กันเรียบร้อย แต่เป็นส่วนเลือก Tables หน้าตามันไม่เหมือนในรูปที่ 7

ลิสต์รายชื่อไฟล์ที่มีสิทธิ์เลือกเป็นนามสกุล .dbf เท่านั้น ด้วยเหตุเพราะ Connect มันได้รับคำสั่งให้ยุ่งเกี่ยวอยู่กับ dBASE ไฟล์ อย่างอื่นห้ามยุ่ง เลือกชนิดดาต้าเบสแบบไหนก็จะมีสิทธิ์เห็นแต่เพียงนามสกุลของดาต้าเบสชนิดนั้น

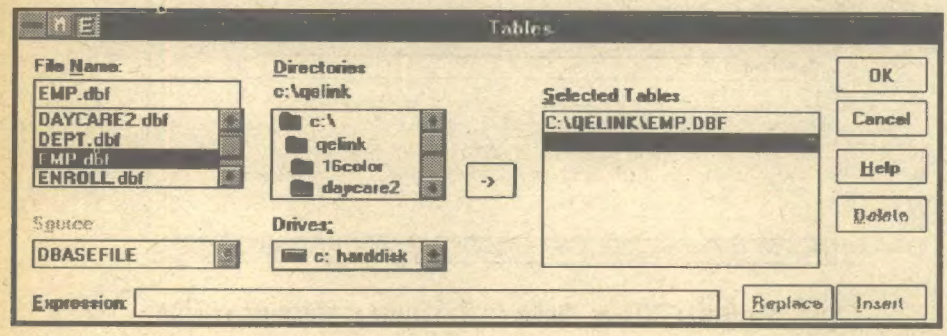
ผมได้เลือกเอาเทเบิลชื่อ EMP.DBF ซึ่งอยู่ในไดเรคทอรี C:\QELINK มาเรียบร้อยด้วยการดับเบิลคลิ๊กที่ชื่อมันตรง ๆ ถ้าหากต้องการเรียกมากกว่าหนึ่งเทเบิลก็ทำได้โดยการดับเบิลคลิ๊กเอาเทเบิลอื่นเข้าไปใส่ไว้ในช่อง Selected Tables อีกเรื่อย ๆ กรณีที่คุณระบุชื่อเทเบิลด้วยวิธีเขียนเองในหน้าต่างพรอพเพอร์ตี้ส์ ก็ต้องใช้วิธีแยกชื่อเทเบิลด้วยคอมม่า (,) นะครับ

ทีนี้มาถึงลำดับที่ว่า หากผมคลิ๊ก OK ไปอะไรจะเกิดขึ้น? ในกรณีที่ผมเลือกเทเบิลเดียวอย่างเช่นที่เลือก EMP.DBF ไปแล้ว จะเห็นหน้าต่าง Query Builder ปรากฏขึ้นทันที

อย่างที่เห็นในปัจจุบัน (ตามรูปที่ 9) จะมีชื่อเทเบิลพร้อมที่อยู่ของมันแสดงอยู่ในช่อง Tables แค่อย่างเดียว ทั้งนี้ก็เพราะเราเพิ่งจะระบุเทเบิลไปรายการเดียวนั่นเอง ถึงขั้นนี้ยังไม่มีอะไรแตกต่างกันระหว่าง Query กับ QGrid

แต่ขั้นต่อไปจะเริ่มฉีกแนวกันแล้วครับ หากตัวเรียกข้อมูลของคุณเป็น QGrid และคุณคลิ๊กปุ่ม OK ลงไปในขณะที่เพิ่งกำหนดชื่อเทเบิลไปอย่างเดียว ผลที่เกิดขึ้นก็คือมันจะกลับไปที่หน้าจอดีไซน์ **โดยถือว่าคุณมอบหมายให้มันเรียกทุกฟิลด์ทุกเรคอร์ดในเทเบิลขึ้นมาแสดง** สังเกตดูได้จากหน้าต่างพรอพเพอร์ตี้ส์ดังรูปที่ 10

การระบุชื่อฟิลด์ที่จะให้ QGrid แสดงนั้นทำได้ที่บรรทัด pQueryExpr เมื่อก่อนเข้าไปเลือกเทเบิลเรายังเห็นบรรทัดนี้มันว่างอยู่จริงไหมครับ แต่พอกลับออกมาปรากฏว่ามีชื่อฟิลด์ทั้งหมดในเทเบิล EMP.DBF มาขึนรออยู่แล้ว นี่แสดงให้เห็นว่าเพียงระบุชื่อเทเบิล ก็เป็นอันเสร็จพิธีกำหนดรายละเอียดอย่างง่ายที่สุด คุณจะได้ฟิลด์ทั้งหมดในเทเบิลขึ้นมาด้วยโดยอัตโนมัติ แต่ว่า...ต้องจำแค่นี้ให้ดีนะครับ คุณต้องไปเลือกเทเบิล




รูปที่ 8 หน้าต่างเลือกเทเบิล

Tables

C:\QELINK\EMP.DBF

รูปที่ 9 ช่องแสดงการเลือกเทเบิลในหน้าต่าง Query Builder

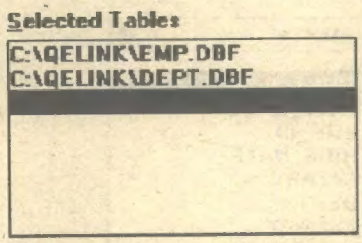
รูปที่ 10 พรอพเพอร์ตี้ pQueryExpr จะมีชื่อฟิลด์ปรากฏขึ้นหลังจากคลิ๊ก OK

ด้วย Query Builder เท่านั้นมันถึงจะทำกิจชนิดนี้ให้เอง ถ้าหากคุณอยู่ที่หน้าจอดีไซน์แล้วพิมพ์ชื่อที่อยู่ของเทเบิลลงไปใน pTables อย่างเดียว มันจะไม่รู้ไม่เห็นอะไรเกี่ยวกับฟิลด์ในเทเบิลนั้นทั้งสิ้น เนื่องจากยังไม่มีการสแกนใด ๆ เกิดขึ้นนั่นเอง

หากตัวเรียกข้อมูลของคุณเป็น Query และคุณคลิ๊กปุ่ม OK ลงไปในขณะที่เพิ่งกำหนดชื่อเทเบิลไปอย่างเดียว ผลที่เกิดขึ้นก็คือมันจะกลับไปที่หน้าจอดีไซน์เหมือนกัน โดยไม่มีการแสดงให้เห็นฟิลด์ใด ๆ ในเทเบิลที่เลือก เนื่องจากไม่มีพรอพเพอร์ตี้ที่ทำหน้าที่นี้อยู่ หากต้องการดู คุณต้องดูตั้งแต่อยู่ใน Query Builder โดยคลิ๊กที่ช่อง Fields ซึ่งเดี๋ยวจะพูดในรายละเอียดเมื่อถึงหัวข้อการเลือกฟิลด์

เมื่อต้องเกี่ยวข้องกับเทเบิลเดียวก็ไม่มีอะไรยุ่งยากอย่างที่เห็นแล้วนั่นแหละครับ แต่ทีนี้ถ้าสมมติว่าคุณต้องการข้อมูลจากเทเบิลอื่นอีกล่ะ เป็นต้นว่าข้อมูลพนักงานว่าด้วย "แผนก" มันไม่มีอยู่ในเทเบิล EMP.DBF แต่ไปมีอยู่ใน DEPT.DBF แน่นอนเราก็ต้องเลือกเทเบิลในหน้าต่าง Tables มากขึ้นโดยใช้กติกาเดิม คือดับเบิลคลิ๊กเลือกที่ช่อง File Name แล้วจะเห็นไปปรากฏอยู่ในช่อง Selected Tables อย่างในรูปที่ 11

สิ่งที่คุณจะต้องทำเมื่อเรียกข้อมูลมาจากต้นแหล่งมากกว่าหนึ่งเทเบิลก็คือ **ระบุว่าจะเชื่อมเทเบิลด้วยฟิลด์อะไรกับฟิลด์อะไร** ไม่เช่นนั้นแล้ว ฟิลด์ทุกฟิลด์ของทุกเทเบิลจะจับคู่กันหมด คุณจะได้ตารางแสดงข้อมูลซ้ำไปซ้ำมามากมายไร้สาระ ยิ่งมากเทเบิล มากฟิลด์เท่าไหร่ยิ่งหนักเข้าไปใหญ่ อยากรู้ว่ามันจะแสดงกี่เรคอร์ดก็ให้เอาจำนวนฟิลด์ของแต่ละเทเบิลมาคูณกันเข้าไป

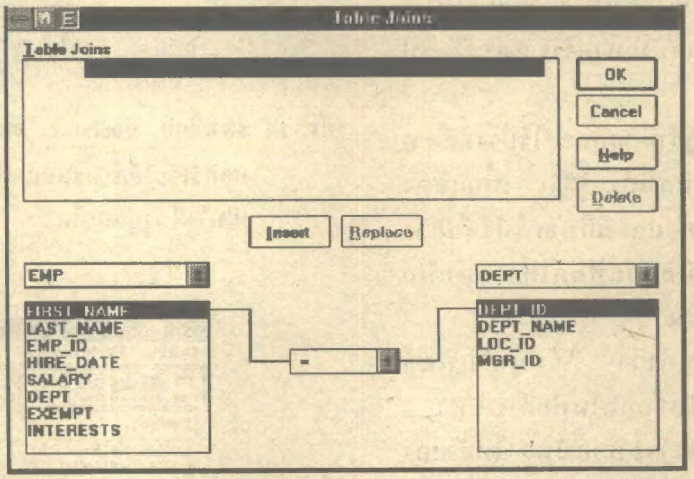



**รูปที่ 11 ช่อง Selected Tables หลังจากเลือก DEPT.DBF เพิ่ม**

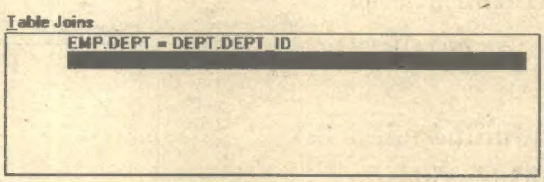
การเชื่อมเทเบิลนั้นจำเป็นจะต้องมีฟิลด์ข้อมูลเหมือนกันอยู่อย่างน้อยหนึ่งฟิลด์ในแต่ละเทเบิล หากปราศจากข้อมูลที่เหมือนกันแล้ว จะไม่มีการแสดงข้อมูลใด ๆ ออกมา เนื่องจากไม่สามารถหาสิ่งที่ตรงตามเงื่อนไขได้ อย่างเช่นคุณให้รหัสพนักงานของเทเบิลที่หนึ่งเท่ากับรหัสพนักงานของเทเบิลที่สอง ถ้าหากเทเบิลแรกมีรหัส 001 กับ 002 แต่เทเบิลที่สองมี 003 กับ 004 ก็แปลว่าไม่มีข้อมูลตรงเงื่อนไข แสดงออกมาไม่ได้ ทั้งที่เชื่อมเทเบิลสำเร็จแล้ว

มาว่าถึงวิธีเชื่อมกัน หลังจากคลิ๊ก OK เมื่อเลือกหลายเทเบิลในหน้าต่าง Tables เรียบร้อย คุณจะเห็นหน้าต่างอีกบานโผล่ขึ้นมา มีชื่อว่า Table Joins ดังในรูปที่ 12

ในรูปแสดงอะไรต่ออะไรค่อนข้างชัดเจนอยู่แล้ว ผมขอให้มาโฟกัสดูตรงจุดสำคัญที่สุดกันก่อน นั่นคือช่องลำดับรายการอันมีชื่อฟิลด์ของสองเทเบิลเรียงกันอยู่ตามโครงสร้างของเทเบิล สังเกตนะครับว่าแถบมืดมันคาดรายการแรกของแต่ละเทเบิลไว้ นี่คือสิ่งที่มันรออยู่เป็นดีฟอลต์ แต่คุณก็เห็นแล้วว่า FIRST_NAME กับ DEPT_ID มันไปด้วยกันไม่ได้แน่นอน เพราะฉะนั้นก็ลองกวาดตาหาฟิลด์ในเทเบิลแรกดูว่ามี




**รูปที่ 12 หน้าต่าง Table Joins ที่ปรากฏขึ้นหลังจากเลือกเทเบิลมากกว่าหนึ่งที่หน้าต่าง Tables**




**รูปที่ 13 การแสดงฟิลด์ที่เชื่อมเทเบิลในช่อง Table Joins หลังจากกดปุ่ม Insert แล้ว**

อะไรที่น่าจะเข้าเค้ามากที่สุด คุณก็จะเห็นชื่อ DEPT ปรากฏอยู่เป็นลำดับที่หก

ก็ขอให้คลิ๊กไปที่ DEPT แล้วคลิ๊กปุ่ม Insert คุณจะเห็นว่ามีการเลือกเกิดขึ้นแล้ว ดังมีการแสดงให้เห็นที่ช่อง Table Joins ที่ส่วนบนสุด ดังในรูปที่ 13

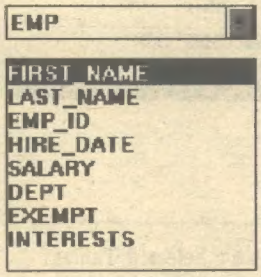
ถ้าหากอ่านรายละเอียดทั้งหมดในหน้าต่าง Table Joins ออกมาเป็นภาษาพูด ก็คงได้เป็นทำนองว่า "ให้แสดงข้อมูลจากเทเบิล **EMP** และ **DEPT** โดยให้แสดงเฉพาะเรคอร์ดที่ข้อมูลในฟิลด์ **DEPT** ของเทเบิล EMP ตรงกับข้อมูลในฟิลด์ **DEPT ID** ของเทเบิล DEPT"

มาวิเคราะห์กันทีละวรรคทีละตอนกันเลย เรารู้ได้อย่างไรว่าให้แสดงข้อมูลจากเทเบิล EMP และ DEPT อันนี้ผมเอามาจากช่องแสดงเทเบิลเหนือช่องแสดงฟิลด์นั่นเองครับ ดูตามในรูปที่ 14

คุณจะดูว่ามีเทเบิลอะไรบ้างที่เลือกไว้แล้วก็ให้คลิ๊กที่ปุ่ม เพื่อแสดงเทเบิลทั้งหมด และนี่ก็บอกให้รู้ด้วยว่าคุณเชื่อมเทเบิลด้วยฟิลด์ใดจากเทเบิลไหนก็ได้ทั้งสิ้น

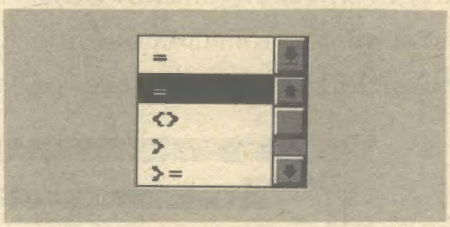
ทีนี้มาดูคำว่า "ข้อมูลในฟิลด์ DEPT ตรงกับข้อมูลในฟิลด์ DEPT_ID" การที่เราจะบอกมันว่าให้คัดเอาเฉพาะเรคอร์ดที่มีข้อมูล "ตรงกัน" นั้น เราต้องระบุชื่อฟิลด์อันเป็นขั้วสัมพันธภาพ และกำหนดให้เครื่องหมายในช่องเลือก operator เป็น = อย่างในรูปที่ 15

เครื่องหมายเท่ากับเป็นตัวเลือกดีฟอลต์อยู่แล้ว ดังนั้นก็ไม่ต้องไปสนใจอะไรอีก แต่หากคุณต้องการใช้เครื่องหมายอื่น เป็นต้นว่า > เพื่อกำหนด




**รูปที่ 14 การแสดงชื่อเทเบิลและชื่อฟิลด์ที่อยู่ในเทเบิล**

เรียกข้อมูลคล้าย ๆ "พนักงานที่เข้าหลังจากวันที่..." อะไรทำนองนี้ ก็คลิ๊กปุ่ม เพื่อเลือกได้ ที่คุณเห็นตัวเลือกมีอยู่เท่านั้นมันไม่ใช่ทั้งหมดนะครับ ยังมีตัวที่เหลืออยู่สองคือ < และ <= อยู่ข้างใต้อีก ใช้วิธีกดปุ่มสโครลบาร์ลงไปเลือกเอา

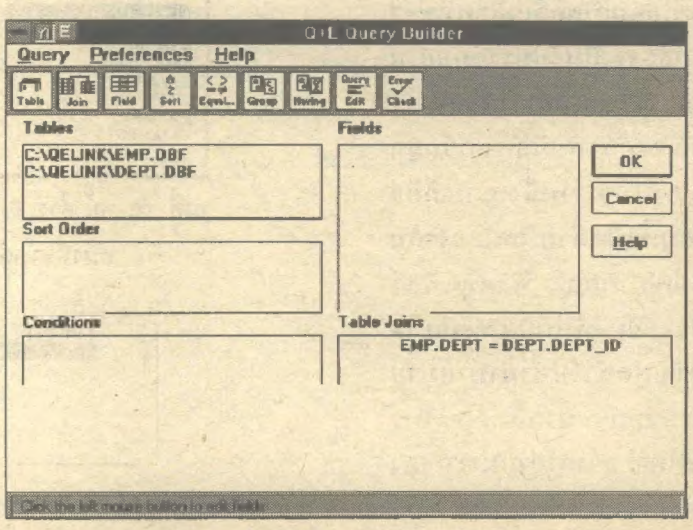



**รูปที่ 15 ช่องเลือก operator อันเป็นตัวบอกว่าจะให้การเชื่อมเทเบิลเป็นไปในรูปแบบใด**

วิเคราะห์กันไปแล้วว่าเราจะต้องทำอะไรบ้างในการเชื่อมเทเบิล ทีนี้ในความเป็นจริงคุณอาจจำเป็นต้องเชื่อมเทเบิลด้วยฟิลด์มากกว่าหนึ่งคู่ ก็ให้เลือกคู่ฟิลด์ที่ต้องการกำหนดเป็นเงื่อนไขข้อใหม่ แล้วคลิ๊กปุ่ม Insert คุณก็จะเห็นว่าช่อง Table Joins มีบรรทัดเงื่อนไขใหม่ปรากฏต่อจากเงื่อนไขบรรทัดแรก โดยมีคำว่า And นำหน้า

เมื่อคุณคลิ๊ก OK ก็จะปรากฏหน้าต่าง Query Builder มาให้เห็นอีก แต่คราวนี้ไม่ได้มีรายละเอียดในการเลือกข้อมูลเป็นชื่อเทเบิลในช่อง Tables อย่างเดียว จะมีรายละเอียดในช่อง Table Joins เพิ่มขึ้นมาด้วย ดังในรูปที่ 16

หากคุณคลิ๊ก OK ก็เป็นอันว่าจบรายละเอียดการเลือก กลับไปตั้งหลักกันที่หน้าจอดีไซน์ แต่ดูแล้วยังมีอีกสองช่องที่ว่างวายอยู่ ดังนั้นจึงยังมีรายละเอียดการเรียกข้อมูลให้พูดถึงอีกสองข้อ คือ การกำหนดชื่อฟิลด์ที่ต้องการให้แสดง (ช่อง Fields) และการกำหนดเงื่อนไขจำเพาะเจาะจง (ช่อง Conditions)




**รูปที่ 16 หน้าต่าง Query Builder หลังจากทำการเชื่อมเทเบิลแล้ว**

# การเลือกฟิลด์

## วิธีเข้าหน้าต่าง Fields

● เมื่ออยู่ในหน้าต่างพรอพเพอร์ตี้ส์ (เฉพาะ QGrid) ดับเบิลคลิกบรรทัด pQueryExpr

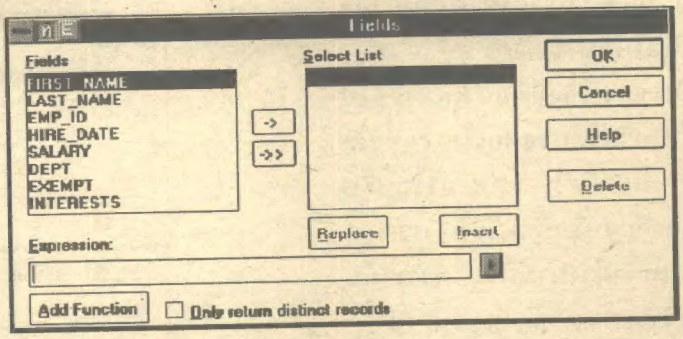
● เมื่ออยู่ในหน้าต่าง Query Builder (ทั้ง Query และ QGrid) ให้คลิกลงในบริเวณที่ว่างของช่อง Fields

ย่อหน้าต่อไปอาจเป็นเรื่องน่างงสักนิด ดังนั้นผมขอทบทวนความรู้เบื้องต้นเกี่ยวกับข้อแตกต่างระหว่าง Query กับ QGrid อีกสักครั้ง คือขึ้นชื่อว่าตัวเรียกข้อมูลแล้ว ไม่ว่าจะเป็น Query หรือ QGrid มัน "เห็น" ทุกสิ่งทุกอย่างในเทเบิลหมดแหละครับ แต่ **คุณต้องเลือกฟิลด์ให้กับ QGrid ทั้งนี้เพื่อระบุว่าต้องการให้มัน "แสดง" ฟิลด์ออกมามากน้อยแค่ไหน** ส่วน Query นั้นไม่ต้องเลือก เพราะยังไง ๆ มันก็ไม่แสดงอะไรให้คุณดูอยู่แล้ว

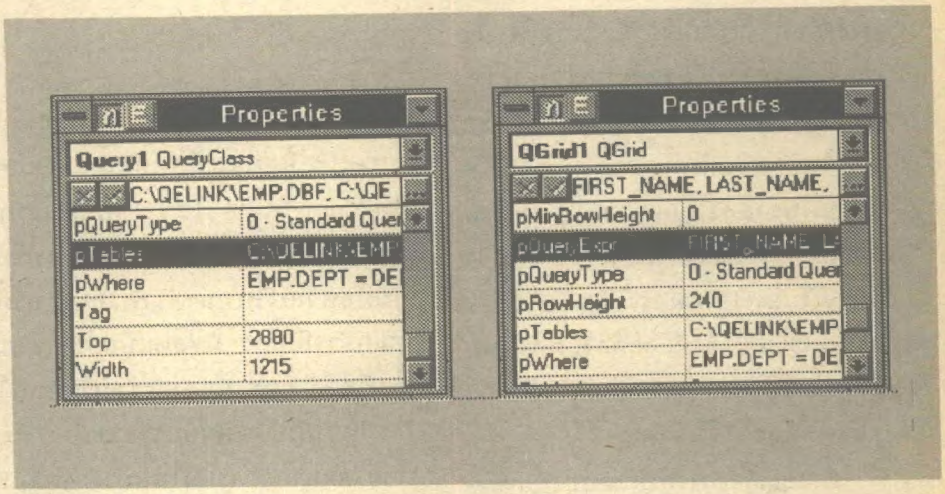
หากอยู่ที่หน้าจอดีไซน์ก็ไม่มีปัญหาอะไร คุณสามารถทราบได้ด้วยตาตัวเองว่าในหน้าต่างพรอพเพอร์ตี้ส์ของ Query นั้นไม่มี pQueryExpr เหมือนในหน้าต่างพรอพเพอร์ตี้ส์ของ QGrid ดังรูปที่ 18 เปรียบเทียบให้ดู

สำหรับพรอพเพอร์ตี้ส์สองตัวหลักคือ pTables กับ pWhere นั้นจะเห็นว่าเหมือนกันทุกประการ เพราะผมดำเนินการอย่างเดียวกันทุกอย่างทั้ง Query และ QGrid โดยที่การเชื่อมเทเบิลจะไปปรากฏอยู่ในพรอพเพอร์ตี้ pWhere (เดี๋ยวจะพูดถึงรายละเอียดในหัวข้อถัดไป)

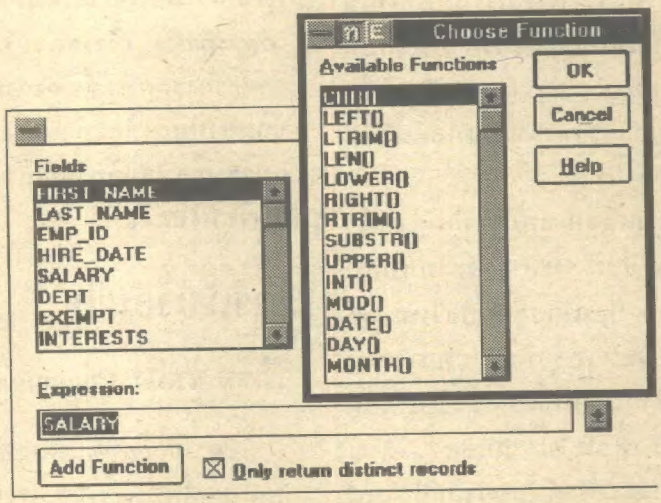
เมื่ออยู่ใน Query Builder นั้นคุณได้เห็นแล้วว่าคุณมีสิทธิ์เข้าไปใน




รูปที่ 17 หน้าต่างเลือกฟิลด์ของ Query Builder




รูปที่ 18 ความแตกต่างกันระหว่างหน้าต่างพรอพเพอร์ตี้ของ Query กับ QGrid




รูปที่ 19 รายละเอียดในการเรียกฟิลด์ที่คุณสามารถปรับแต่งได้ในหน้าต่าง Fields

หน้าต่าง Fields ได้ทั้ง ๆ ที่กำลังเลือกเทเบิลให้กับ Query อยู่

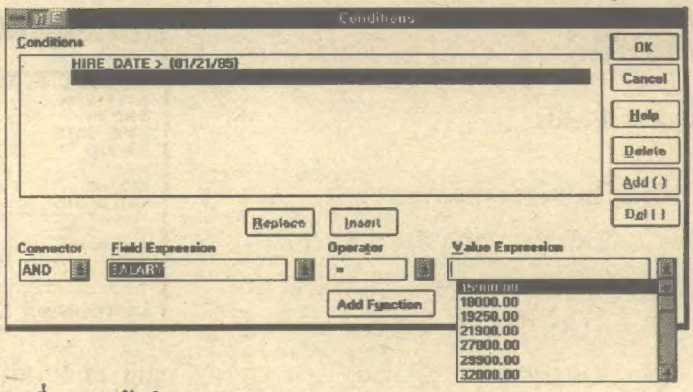
เรื่องของเรื่องก็คือสั่งเอาหรือไม่เอาฟิลด์ไหนไม่ได้หรอกครับ (แม้เมื่อลองดูจะเหมือนได้) แต่สามารถปรับแต่งรายละเอียดหรือเงื่อนไขในการเรียกฟิลด์ได้ในหน้าต่างบานนั้น ดังจะเห็นจากที่ผมจับภาพมาให้ดูในรูปที่ 19

เมื่อคุณกดปุ่ม Add Function จะเห็นมีหน้าต่างบานเล็กลิสต์รายชื่อฟังก์ชันที่คุณมีสิทธิ์ใช้ขึ้นมาให้เลือก อย่างเช่นที่รู้จักกันดีคือ Left กับ Right นอกจากนั้นยังมี CheckBox ให้เลือกว่าจะแสดงเรกอร์ดที่ซ้ำกันทุกฟิลด์หรือไม่ ในการเริ่มต้นนี้ผมจะไม่นำคุณไปพัวพันกับรายละเอียดต่าง ๆ เหล่านี้มากนัก ให้คุณได้จำแม่น ๆ ก็พอว่า Query ไม่มีพรอพเพอร์ตี้ pQueryExpr ต่างกับ QGrid ซึ่งมีและต้องถูกกำหนดโดยคุณหรือ Query Builder ด้วยว่าจะนำฟิลด์อะไรขึ้นมาแสดงบ้าง

หากคุณต้องการกำหนดค่า pQueryExpr ให้กับ QGrid โดยไม่ผ่าน Query Builder ก็มีอีกวิธีคือ เขียนลงไปตรง ๆ ที่บรรทัด pQueryExpr ในหน้าต่างพรอพเพอร์ตี้ส์ได้ หลักการก็คือ หากมีมากกว่าหนึ่งฟิลด์ คุณต้องแยกพวกมันด้วยเครื่องหมาย , ส่วนเรื่องตัวพิมพ์เล็กพิมพ์ใหญ่และวรรคระหว่างฟิลด์นั้นไม่ใช่เรื่องสำคัญอะไร

ถ้าหากเลือกเทเบิลมากกว่าหนึ่งทั้ง Query และ QGrid จะไม่สับสนงุนงงอะไรว่าฟิลด์ใดอยู่เทเบิลไหน แต่มันมีอยู่กรณีหนึ่งที่ชื่อฟิลด์ระหว่างสองเทเบิลมันซ้ำกัน อันนี้แหละครับจะเกิดความงงขึ้น เพราะไม่ "แน่ชัด" ว่าคุณต้องการให้แสดงฟิลด์นั้นของเทเบิลไหน วิธีแก้ปัญหาก็ง่ายนิดเดียว คือคุณต้องระบุชื่อเทเบิลนั้นนำหน้า ต่อด้วยจุดและตามด้วยชื่อฟิลด์ อันนี้หากคุณเลือกหลายเทเบิลและฟิลด์ผ่าน Query Builder มันก็จะใส่ชื่อเทเบิลข้างหน้าฟิลด์ทุกฟิลด์ให้ แต่หากคุณต้องการเขียนกับมือตัวเองในหน้าต่างพรอพเพอร์ตี้ส์ คุณก็อาจเลือกระบุชื่อเทเบิลเฉพาะหน้าชื่อฟิลด์ที่ชื่อซ้ำกันระหว่างเทเบิลก็ได้ (ในทางปฏิบัติผมชอบระบุเทเบิล/ฟิลด์ด้วยโค้ดมากกว่าระบุในหน้าต่างพรอพเพอร์ตี้ส์ เนื่องจากเราสามารถเห็นรายละเอียดการเรียกข้อมูลอย่างชัดเจนขณะก่อนปฏิบัติการเรียกจริง เมื่อจะอ้างอิงหรือดัดแปลงแก้ไขเพิ่มลดอย่างไรก็ทำ copy/paste ได้ง่ายด้วย)

เอาล่ะครับ รู้เรื่องการกำหนดเรียกข้อมูลไปสองข้อแล้วคือระบบชื่อเทเบิลกับระบุชื่อฟิลด์ต่อไปก็เป็นเรื่องของการตั้งเงื่อนไขต่าง ๆ




รูปที่ 20 หน้าต่าง Conditions

## การตั้งเงื่อนไข

### วิธีเข้าหน้าต่าง Conditions

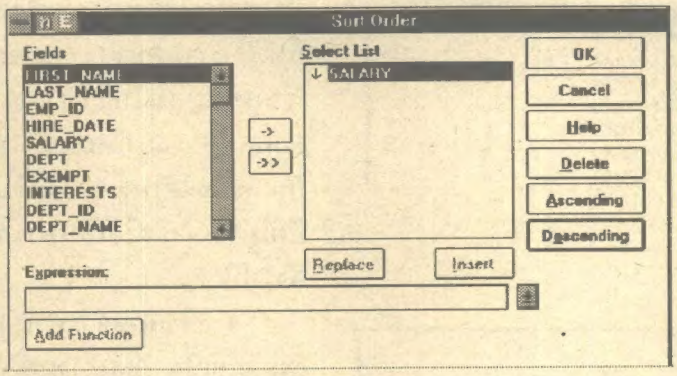
- เมื่ออยู่ที่หน้าต่างพรอพเพอร์ตี้ส์ ดับเบิลคลิ๊กที่บรรทัด pWhere
- เมื่ออยู่ที่หน้าต่าง Query Builder ให้คลิ๊กในบริเวณที่ว่างของช่อง Conditions

การตั้งเงื่อนไขนั้นเป็นเรื่องของพรอพเพอร์ตี้ pWhere เนื้อหาของเงื่อนไขมักเกี่ยวข้องกับค่าของฟิลด์ ว่าจะเท่ากับหรือมากกว่าน้อยกว่าปริมาณนั้นปริมาณนี้ เช่นขอให้เรียกเฉพาะข้อมูลพนักงานที่เงินเดือนมากกว่าหนึ่งหมื่นบาทเป็นต้น

ในทางปฏิบัติ ถ้าคุณใส่ค่าเองที่หน้าต่างพรอพเพอร์ตี้ส์ ก็หมายความว่าคุณรู้อะไรดี ๆ เกี่ยวกับข้อมูลที่อยู่ในฟิลด์นั้น ๆ อยู่แล้ว อย่างเช่นคุณรู้แน่ว่ามีพนักงานเงินเดือนมากกว่าหนึ่งหมื่น คุณถึงใส่ลงไปได้ถูกว่าให้เรียกเฉพาะพนักงานที่เงินเดือนมากกว่าหมื่นขึ้นไปมาแสดง

แต่ในกรณีที่ไม่รู้ก็มีบริการพิเศษให้ คุณจะเห็นว่าหน้าต่าง Conditions ช่วยได้มากทีเดียว ทั้งยังเป็นการประกันความผิดพลาดให้ร้อยเปอร์เซ็นต์ เนื่องจากมันจะลิสต์ค่าของข้อมูลทั้งหมดในฟิลด์นั้นมาให้คุณเลือกโดยตรง ขอให้ดูรูปที่ 20 ตรงช่อง Value Expression

ขอให้อ่านตามกันเลยครับ จากรูปที่ 20 นี้ คุณจะเห็นในช่อง Conditions ว่ามีการตั้งเงื่อนไขไว้แล้วข้อหนึ่ง




**รูปที่ 21 หน้าต่าง Sort Order**

คือ HIRE_DATE>{01/21/85} หมายถึงให้ดึงข้อมูลพนักงานที่เข้าทำงานหลังจาก วันที่ 21 มกราคม ปี พ.ศ. 2528 สังเกตนะครับว่ารูปแบบการให้ค่าข้อมูลชนิด Date ต้องใส่เครื่องหมาย { } ขอบอกไว้ ณ ที่นี้เลยว่าถ้าค่าเป็น Character ต้องมีเครื่องหมาย ' ' และถ้าค่าเป็น Numeric ก็ไม่ต้องห่อด้วยอะไรเลย ใส่ตัวเลขลงไปเฉย ๆ ต้องระวังให้เครื่องหมายตรงตามชนิดข้อมูลด้วยนะครับ ไม่อย่างนั้นมันจะฟ้อง Incompatible Data Type...

และที่ส่วนล่างของหน้าต่างกำลังตั้งเงื่อนไขอีกข้อหนึ่ง คือเงินเดือนพนักงาน ในลิสต์ของช่อง Value Expression เป็นการดึงเอาข้อมูลพนักงานฟิลด์ SALARY ทั้งหมดมาให้เลือกว่าคุณต้องการให้แสดงเฉพาะผู้ที่มีเงินเดือนเท่ากับ น้อยกว่า หรือมากกว่าจำนวนเท่าใด อาจเป็นตัวเลขที่ไม่อยู่ในลิสต์ก็ได้นะครับ เช่นคุณอยากดูพนักงานที่เงินเดือนมากกว่าหนึ่งหมื่นก็ใส่เลข 10000 แล้วเลือกเครื่องหมาย (operator) เสียใหม่ให้เป็น ">" จากนั้นคลิกปุ่ม Insert ก็เป็นอันเรียบร้อย จะเห็นเงื่อนไขอีกข้อหนึ่งปรากฏบนช่อง Conditions เป็นข้อความว่า

And SALARY > 10000

เสร็จแล้วคราวนี้เมื่อดูเงื่อนไขทั้งหมด ก็จะเป็น

HIRE_DATE > {01/21/85} And SALARY > 10000

สังเกตที่คำว่า And ซึ่งเป็นตัวเชื่อมเงื่อนไขซึ่งมาจากช่อง Connector ที่ช่องนี้เมื่อแรกก่อนการตั้งเงื่อนไขใด ๆ คุณมีตัวเลือกเพียงสองตัวคือความว่างเปล่ากับ NOT แต่ถ้าหากตั้งเงื่อนไขไปแล้วอย่างน้อยหนึ่งข้อ คราวนี้ Connector จะเป็นความว่างเปล่าไม่ได้อีกต่อไป อย่างน้อยต้องมี AND มาเป็นตัวเชื่อมเงื่อนไข และคุณยังมีสิทธิ์เลือกตัวเชื่อมได้อีกสามคือ OR, AND NOT, และ OR NOT

ถ้าอ่านเงื่อนไขที่ตั้งไว้แล้ว ก็อาจถอดเป็นภาษาพูดได้ว่า ให้แสดงข้อมูลพนักงานที่เข้าบริษัทหลัง 21 มกราคม 2528 และมีเงินเดือนมากกว่าหนึ่งหมื่นบาทขึ้นไป

## การลำดับข้อมูล

เหลืออีกข้อสุดท้ายที่จะพูดเกี่ยวกับ Query Builder นั่นคือการเรียงลำดับข้อมูล งานนี้คุณยังเกี่ยวข้องกับพรอพเพอร์ตี้ pWhere เหมือนกับการตั้งเงื่อนไข แต่จะต้องใช้หน้าต่างอีกบานหนึ่งต่างหาก อันมีชื่อว่า Sort Order ดังที่เห็นในรูปที่ 21

คุณสามารถเข้าหน้าต่างนี้ได้ด้วยการคลิกช่อง Sort Order ที่หน้าต่าง Query Builder โดยตรง

การเลือกก็ง่ายนิดเดียวครับ สมมติว่าผมอยากให้ข้อมูลทั้งหมดถูกลำดับด้วย SALARY ผมก็ดับเบิลคลิก SALARY ที่ช่อง Fields ก็จะปรากฏว่า SALARY มันไปอยู่ที่ช่อง Select List โดยมีลูกศรชี้ขึ้นนำหน้า (หมายถึงให้เรียงจากน้อยไปหามาก) เท่านี้เป็นอันจบกระบวนการกำหนดลำดับ

แต่จากรูปที่ 21 คุณจะเห็นว่าลูกศรนำหน้า SALARY นั้นชี้ลง (เรียงจากมากไปหาน้อย) ที่เป็นดังนั้นเพราะผมคลิก SALARY ที่ช่อง Select List ซึ่งเมื่อทำดังนั้นจะเห็นว่าปุ่ม Delete, Ascending กับ Descending มันแอคตีฟขึ้นมา สำหรับ Ascending นั้นหมายถึงเรียงจากน้อยไปหามากซึ่ง

| | FIRST_NAME | LAST_NAME | EMP_ID | HIRE_DATE | SALARY | DEPT | EXEMPT | INTERESTS |
|---|---|---|---|---|---|---|---|---|
| 1 | Kim | Aulich | E10001 | 7/30/85 | 57000.00 | D190 | 1 | |
| 2 | John | Rappl | E21437 | 7/15/87 | 47000.00 | D050 | 1 | Sports;~~ |
| 3 | Tim | Sampair | E03033 | 12/2/87 | 27000.00 | D101 | 1 | |
| 4 | Nathan | Adams | E41298 | 2/15/88 | 21900.00 | D050 | 0 | |
| 5 | David | Motsinger | E27002 | 5/5/85 | 19250.00 | D202 | 0 | |
| 6 | Adam | Smith | E63535 | 1/15/88 | 18000.00 | D202 | 0 | |
| 7 | Richard | Poltes | E43128 | 4/12/86 | 15900.00 | D101 | 0 | |

**รูปที่ 22 การแสดงข้อมูลของ QGrid เมื่อทำการ Do Query**

```
QGrid1.pTables = "\QELINK\EMP.DBF, \QELINK\
    DEPT.DBF"
QGrid1.pQueryExpr = "FIRST_NAME, LAST_NAME,
    EMP_ID, HIRE_DATE, SALARY, DEPT,
    EXEMPT, INTERESTS, DEPT_ID, DEPT_NAME,
    LOC_ID, MGR_ID"
QGrid1.pWhere = "EMP.DEPT = DEPT.DEPT_ID
    And HIRE_DATE > {01/21/85} AND SALARY >
    10000 ORDER BY SALARY DESC"

res% = fDoQuery(QGrid1)
```

**รูปที่ 23**

เป็นค่าดีฟอลต์อยู่แล้ว ส่วน Descending หมายถึงกำหนดให้เรียงจากมากไปหาน้อย ซึ่งเมื่อผมคลิ๊กปุ่มนี้ ก็จะปรากฏลูกศรชี้ลงดังที่คุณได้เห็น

เมื่อคลิ๊ก OK มาที่หน้าต่าง Query Builder คุณจะเห็นทุกช่องเต็มไปด้วยรายละเอียดการเรียกข้อมูล นับจาก Tables, Fields, Conditions มาจนถึง Sort Order หากกลับไปที่หน้าจอดีไซน์ ก็จะเห็นบรรทัด pWhere มีคู่ฟิลด์เชื่อมเทเบิลนำหน้า แล้วปิดท้ายด้วยตัวกำหนดลำดับข้อมูล Order By ... (ซึ่งเป็นกฎของ SQL ว่า Order By จะต้องอยู่ส่วนท้ายสุดของ clause)

ที่จุดนี้คงหมดปัญหาเกี่ยวกับการใช้ Query Builder เบื้องต้นกันเรียบร้อย ทีนี้มาพูดกันถึงเรื่องการแสดงข้อมูลล่ะครับ มันมีความแตกต่างอย่างมากระหว่างการเรียกข้อมูลด้วย Query กับ QGrid

ถ้าหากคุณเรียกข้อมูลด้วย QGrid ก็เป็นอันว่าจะได้เห็นตัวข้อมูลที่อยู่ในขอบข่ายของการระบุเทเบิล ฟิลด์ และเงื่อนไขทั้งหมดโดยไม่ต้องต่อเติมอะไร หากว่ากันอย่างง่ายที่สุด คุณเพียงแต่สั่ง

```
res% = fDoQuery(QGrid1)
```

fDoQuery ( ) เป็นฟังก์ชันแรกที่ต้องใช้ในการติดต่อกับดาต้าเบส หากปราศจากฟังก์ชันนี้แล้วฟังก์ชันอื่นใด ๆ ก็หมดความหมาย คุณจะเอาบรรทัดเรียกข้อมูลนี้ไปไว้ที่ไหนก็ได้ เช่น Form_Load ก็หมายถึงให้เรียกข้อมูลกันเลยตั้งแต่โหลดฟอร์มขึ้นมา เท่านี้ก็เป็นอันว่าจะได้เห็นการแสดงข้อมูลดังรูปที่ 22

จะทดลองดูด้วยตัวเองก็ได้นะครับ เพราะเทเบิล EMP.DBF กับ DEPT.DBF นั้นเป็นตัวอย่างมีจริงที่ติดมากับ Q+E Multilink/VB หากใครต้องการกำหนดรายละเอียดการเรียกข้อมูลไว้ในโค้ด ก็อาจทำได้ก่อนทำการ Do Query ผมขอยกตัวอย่างเดิมมาสาธิตดังรูปที่ 23

ธรรมชาติของ QGrid จะนำชื่อฟิลด์มาเป็นเฮดดิ้ง แล้วลำดับข้อมูลด้วยเลข 1,2,3... ขอบอกไว้ในที่นี้เลยว่าไม่มีทางใดที่จะไปเปลี่ยนสิ่งต่าง ๆ เหล่านี้ ในทางปฏิบัติเราคงไม่นิยมนำมันมาแสดงจริงทางที่เป็นไปได้คือเลี่ยงไปใช้ Grid มาถ่ายทอดข้อมูล เพราะเราสามารถกำหนดเฮดดิ้งและ Item (คอลัมน์แรกแนว Y) ให้กับ Grid ได้

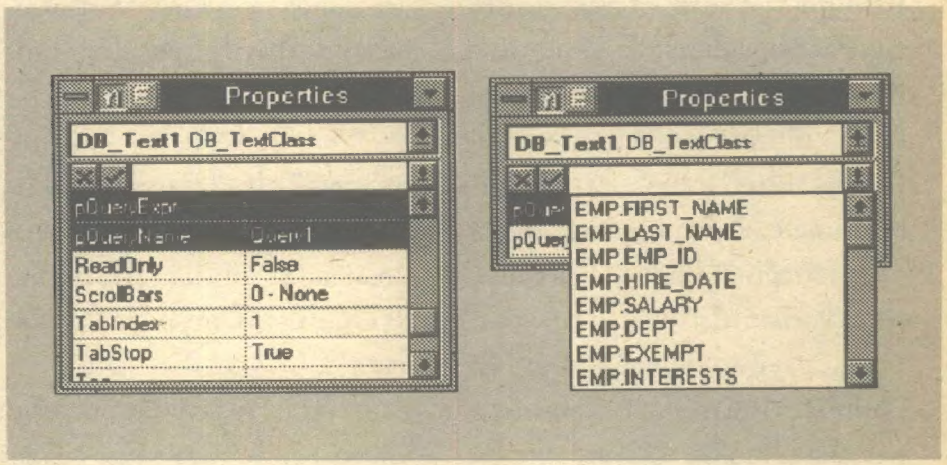
รูปที่ 24 DB_TextBox หรือที่เรียกกันง่าย ๆ ว่า DB_Text

อีกทั้งยังตกแต่งขัดเกลาการนำเสนออื่น ๆ ได้มากกว่า เป็นต้นว่าคิดค่ารวมแถวแนวได้ ใส่ค่าผ่านพรอพเพอร์ตี้ Text เช่น Grid1.Text = Temp1 + Temp2 ซึ่ง Temp1 และ Temp2 อาจหมายถึงข้อมูล ณ เซลใด ๆ ของ QGrid ซึ่งหากเป็นกรณีของตัว QGrid เอง พรอพเพอร์ตี้ Text จะเป็น Read-Only คุณไม่สามารถส่งค่าใด ๆ ทางโค้ด

ข้อดีของ QGrid จริง ๆ แล้วจึงไม่ใช่ความสามารถที่จะแสดงข้อมูลได้ในตัวเอง แต่เป็นความสามารถในการดึงฟิลด์มาได้หลายฟิลด์พร้อมกันต่างหาก เพราะเมื่อใช้จริง ส่วนมากคุณจะพบว่าจำเป็นต้อง "ซ่อน" มันไว้แล้วอาศัย Grid หรือคอนโทรลตัวอื่นที่เข้าขากันได้เป็นต้นว่า ListBox มาแสดงข้อมูลแทน

ซ้ำร้าย QGrid ยังมีข้อเสียที่ผมยังไม่ได้กล่าวถึง นั่นคือมันดึงข้อมูลได้อย่างเดียว แต่ผู้ใช้จะเปลี่ยนแปลงแก้ไขอะไรผ่านมันไม่ได้เลย หากต้องการจะเพิ่ม, ลบ, แก้ไขข้อมูลในเทเบิลจริง ๆ คุณต้องใช้ Query ครับ

การใช้ Query นั้นต้องอาศัยคอนโทรลอื่นมาช่วยแสดงแน่นอน ในเมื่อไม่มีส่วนใดของมันทำหน้าที่นี้ได้ และก็ต้องเป็นคอนโทรลที่มากับ Q+E Multilink/VB เท่านั้น Query และคอนโทรลจึงจะรู้จักกันเป็นการ "ภายใน"




รูปที่ 25 พรอพเพอร์ตี้ที่จะทำให้ DB_TextBox แสดงข้อมูลจาก Query ได้

ก่อนอื่นขอแนะนำให้รู้จัก DB_TextBox

DB_TextBox นั้นถ้าคิดง่าย ๆ ก็คือ TextBox อันเป็นคอนโทรลมาตรฐานสำหรับการอินพุตข้อมูลของวิชวลเบสิกนั่นเอง เมื่อสร้างขึ้นมาทีแรกจะมีหน้าตาดังรูปที่ 24

คอนโทรลตัวนี้และตัวอื่นที่รู้จักกับ Query เป็นการ "ภายใน" นั้นจะมีพรอพเพอร์ตี้สำหลักอยู่สองตัวคือ pQueryExpr และ pQueryName ดังแสดงในรูปที่ 25

ก่อนอื่นคุณต้องระบุชื่อ Query ที่จะทำการติดต่อด้วย (ซึ่งเป็นไปได้ทั้ง Query และ QGrid) จากนั้นคุณต้องระบุว่าจะให้มันแสดงฟิลด์ไหน เพราะมันแสดงได้แค่ฟิลด์เดียว ก็ด้วย pQuery-Expr ที่หายไปจากหน้าต่างพรอพเพอร์ตี้ของ Query นั่นแหละครับ หลบมาอยู่ที่นี่เอง

หาก Query1 ซึ่งคุณระบุว่าจะให้เป็นผู้ส่งข้อมูลมา มีเทเบิลอันเป็นเป้าหมายเรียกข้อมูลแน่นอนแล้ว เมื่อคลิ๊กปุ่ม ▼ ข้างช่องอีดิตข้อมูล ก็จะ

ปรากฏลิสต์รายชื่อฟิลด์ทั้งหมดซึ่งอยู่ในทุกเทเบิลที่คุณเลือกไว้ ให้คลิ๊กชื่อฟิลด์ที่ต้องการให้ DB__TextBox มันแสดง เป็นต้นว่า FIRST__NAME (ในที่นี้จะเห็นชื่อเทเบิล EMP นำหน้าและคั่นด้วยจุด เห็นแค่นี้คุณคงรู้ได้ทันทีว่าหมายถึงมีเทเบิลที่เลือกไว้ใน Query มากกว่าหนึ่งขึ้นไป)

เมื่อจะเรียกข้อมูลจริงก็ใช้ฟังก์ชันเดิมก็คือ fDoQuery( ) โดยเปลี่ยนพารามิเตอร์จากชื่อ QGrid1 มาเป็น Query1 (ซึ่งเป็นชื่อดีฟอลต์แรกของ Query) เสียแทน เท่านี้เมื่อทำการรันคุณก็จะเห็นข้อมูลในฟิลด์ FIRST__NAME เรคอร์ดแรกปรากฏอยู่ใน DB__Text

มาพูดถึงทางเลือกอื่นในการเรียกใช้ฟังก์ชัน fDoQuery( ) กัน นอกจากคุณจะทำได้ในโค้ดแล้วยังสามารถใช้ปุ่มพิเศษสั่งมันทำงานให้คุณอย่างง่าย ๆ คือสรุปแล้วคุณไม่จำเป็นต้องเขียนโค้ดแม้แต่บรรทัดเดียว

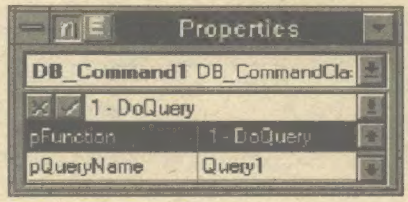
เครื่องทุ่นแรงพิเศษนี้มีชื่อตามชนิดคอนโทรลว่า DB__Command-Button มันก็หลักเดียวกับ Command-Button นั่นแหละครับ แต่มีข้อแตกต่างเพิ่มเติมที่เยี่ยมยอดคือมีพรอพเพอร์ตี้ตัวหนึ่งบรรจุฟังก์ชันเกี่ยวกับดาต้าเบสไว้ นั่นหมายความว่าถ้าค่าพรอพเพอร์ตี้ตัวนี้ไม่เป็น None ล่ะก็ เมื่อคุณคลิ๊กปุ่มเมื่อไหร่ จะมีการทำงานทางดาต้าเบสอย่างใดอย่างหนึ่งขึ้นทันที โดยคุณไม่ต้องไปเขียนโค้ดสั่งมันที่โพรซีเดอร์ DB__Command1__Click( ) เหมือนอย่าง CommandButton

พรอพเพอร์ตี้ที่ว่านี้ชื่อ pFunction ครับ โดยปกติมันจะเป็น 0-None หมายความว่าคลิ๊กไปก็ไม่ทำอะไรให้ซักอย่าง

แต่ถ้าคุณเลือกค่า 1-fDoQuery อย่างที่เห็นในรูปที่ 26 ล่ะก็ แค่นี้ได้เรื่องแล้วล่ะครับ ลบสเตทเม้นท์ res% = fDoQuery(Query1) ออกไปจากหน้าต่างอีดิตโค้ดได้เลย คราวนี้เมื่อคุณรันโปรแกรมใหม่ แล้วคลิ๊กปุ่ม DB__Command1 ทีหนึ่ง ก็จะเห็นข้อมูลปรากฏเหมือนเมื่อครั้งเรียกด้วยโค้ดทุกประการ

หลักการทำงานด้วยปุ่มดาต้าเบสนี้คือหนึ่งปุ่มต่อหนึ่งฟังก์ชัน ถ้าอยากได้ปุ่มที่เรียกฟังก์ชัน Next ก็ต้องสร้างขึ้นมาอีกปุ่มหนึ่งสำหรับฟังก์ชัน Next โดยเฉพาะ ถ้าอยากได้ฟังก์ชัน New, Delete, Update หรืออะไรอย่างอื่นที่เหลือก็สร้างขึ้นเป็นปุ่ม ๆ ไป และมากำหนดเอาที่ pFunction นี้ว่าจะให้มันทำงานอะไรโดยเฉพาะ

โปรดระลึกไว้ว่าคอนโทรลที่เกี่ยวกับดาต้าเบสนี้ทำหน้าที่เพียงรับหรือจัดการข้อมูลใน Query ทีละฟิลด์และทีละเรคอร์ด การเปลี่ยนเรคอร์ดด้วยฟังก์ชัน Next หรือ Previous ก็ดี การแก้ไข




**รูปที่ 26 พรอพเพอร์ตี้หลักของ DB__CommandClass ที่จะทำให้การติดต่อกับดาต้าเบสเป็นไปได้โดยไม่ต้องเขียนโค้ดแม้แต่บรรทัดเดียว**

เพิ่มเติมข้อมูลด้วย New, Update หรือ Delete ก็ดี ล้วนแล้วแต่เป็นการกระทำกับ Query ทั้งสิ้น เหล่าคอนโทรลบริวารไม่มีส่วนรู้เรื่องอะไรด้วยเลย

ฉะนั้นหากคุณมี DB__Text Box หลายตัวเพื่อแสดง/รับข้อมูลหลายฟิลด์ ก็อย่าแปลกใจว่าเมื่อคลิ๊กปุ่ม DB__CommandButton ตัวเดียวแล้วข้อมูลเปลี่ยนแปลงไปหมด คอนโทรลเหล่านั้นไม่ได้เป็นผู้ทำหน้าที่เปลี่ยนแปลงอะไรให้หรอกครับ Query ต่างหากที่เป็นผู้ปฏิบัติ

สำหรับการใช้ฟังก์ชันทางดาต้าเบสอื่นนอกเหนือไปจาก fDoQuery ( ) หลายตัวมีจำนวนพารามิเตอร์มากกว่าหนึ่ง อย่างเช่น fDelete ( ) นี่ต้องระบุด้วยว่าจะให้ลบข้อมูลที่เรคอร์ดไหน หากคุณใช้วิธีเขียนโค้ดก็ต้องระบุอินเด็กซ์ของ Row ต่อท้ายชื่อ Query ไว้ด้วย เช่น

```
res% = fDelete (Query1, 0)
```

โดย 0 เป็นเรคอร์ดปัจจุบันที่กำลังแสดงให้เห็น

ยังมีคอนโทรลเกี่ยวกับดาต้าเบสอีกตัวซึ่งผมคิดว่ามีประโยชน์มาก แต่อาจทำความขลุกขลักให้กับผู้เริ่มใช้พอควร นั่นคือ DB__ComboBox

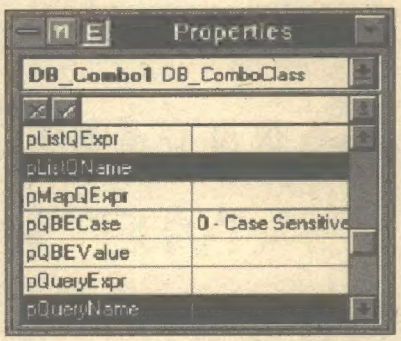
คอนโทรลตัวนี้ถ้าคิดอย่าง user ก็เป็นคอนโทรลเพียงตัวเดียว คือลูกผสมระหว่าง DB__TextBox กับ DB__ListBox แต่หากคิดอย่างโปรแกรมเมอร์แล้ว มันเป็นคอนโทรลสองตัว เพียงแต่นำ DB__TextBox กับ DB__ListBox มาประกบกันหลอกตาเท่านั้น

ที่กล่าวอย่างนี้ดูจากตรงไหน ดูตรงที่คุณต้องใช้ Query ถึงสองตัว ตัวหนึ่งเอาไว้ส่งข้อมูลมาให้ส่วนที่เป็น DB__TextBox อีกตัวหนึ่งเอาไว้ส่งมาให้ส่วนที่เป็น DB__ListBox ขอให้ดูรูปที่ 27

สำหรับ pQueryName นั้นมีไว้อ้าง Query ให้ส่วนที่เป็น DB__TextBox โดยระบุฟิลด์ที่จะให้แสดงด้วย pQueryExpr เหมือนที่คุณเคยทำใน DB__TextBox ทุกประการ

แต่สำหรับ pListName นั้นมีไว้อ้าง Query ให้กับส่วนที่เป็น DB__ListBox โดยระบุฟิลด์ที่จะให้แสดงด้วย pListQExpr ขอให้ทราบไว้ในที่นี้ว่ามันจะดึงเอาข้อมูลในฟิลด์นั้นทุกเรคอร์ดออกมาแสดง โดยจะเห็นได้ต่อเมื่อกดปุ่ม เพื่อขอดูลิสต์ข้อมูลเท่านั้น หากคุณไม่ระบุ pListName หรือ pListQExpr คุณก็จะได้เห็นข้อมูลที่อยู่ในส่วนของ DB__TextBox เท่านั้น

สังเกตนะครับว่าโดยหลักการอย่างนี้ การนำเสนอข้อมูลของ DB__ComboBox อาจแยกส่วนแสดงต่างเทเบิลหรือต่างฟิลด์ ทั้งที่อยู่ในคอนโทรลตัวเดียวกัน คิดแล้วก็อาจใช้ประโยชน์ได้ในบางกรณีแต่คิดอีกทีก็ต้องเปลืองแรงระบุเทเบิลกับฟิลด์สองหน ไม่ทำให้ครบถ้วนก็ไม่ได้เสียด้วย




**รูปที่ 27 พรอพเพอร์ตี้ซึ่งอ้างแหล่งที่มาข้อมูลถึงสองตัวของ DB__ComboClass**

## การสร้าง/แก้ไขเทเบิล

ขอกล่าวอย่างรวบรัดเลยนะครับว่าคุณต้องเข้าไปใน Database Manager ของ Q+E Multilink/VB แล้วทำการดีไฟล์เทเบิล ถ้าต้องการสร้างใหม่เลยให้คลิ๊กปุ่ม จะมีหน้าต่างดีไฟน์ว่าง ๆ โผล่ขึ้นมาให้คุณทำการออกแบบโครงสร้างเทเบิล โดยมีการให้กำหนด

1. ชื่อฟิลด์
2. ชนิดของข้อมูล
3. ความกว้างของฟิลด์ (กี่อักษร)
4. จำนวนทศนิยม (กรณีข้อมูลชนิดตัวเลข)

แต่ถ้าหากคุณต้องการจะแก้ไขโครงสร้างเทเบิลเก่า ก็ให้คลิ๊กปุ่ม จะมีหน้าต่างเลือกเทเบิลซึ่งสร้างไว้แล้วปรากฏเป็นลิสต์รายชื่อ คุณก็ดับเบิลคลิ๊กเอามาทำการแก้ไขตามต้องการ

เมื่อสร้างหรือแก้ไขโครงสร้างเทเบิลเสร็จ ก็ให้เซฟลงดิสค์ด้วยปุ่ม และหากต้องการดูข้อมูลที่อยู่ในเทเบิล ก็ให้คลิ๊กปุ่ม รายละเอียดอื่นที่เหลือคงง่ายที่คุณจะศึกษาเองแล้ว

การใส่ข้อมูลนั้นทำไม่ได้ที่ Database Manager ของ Q+E หรอกนะครับ คุณต้องเขียนโปรแกรมขึ้นมาเอง หรือไม่ก็มีทางเลือกอีกทางที่ง่ายขึ้น คือใช้ Data Control ของวิชวลเบสิก ซึ่งจะนำมาเสนอในตอนต่อไป

## ก่อนจาก

คงไม่ไหวหรอกครับที่ผมจะกล่าวถึงรายละเอียดทั้งหมดของ Q+E Multilink/VB ในที่นี้ แต่ก็หวังว่าเท่าที่กล่าวมาแล้วคงเพียงพอที่จะพาคุณออกจากจุดสตาร์ตได้ในเวลาอันรวดเร็ว ฉบับหน้าผมจะนำการติดต่อกับดาต้าเบสผ่าน Data Control ของวิชวลเบสิกมาให้คุณได้เปรียบเทียบกับ Q+E Multilink/VB ดูครับ สวัสดี